

1. Implementar la clase **Nota** para cumplir con la siguiente interfaz:

```
class Nota {  
    Nota(int valorInicial){...}  
  
    int obtenerValor(){...}  
  
    boolean aprobado(){...}  
  
    boolean desaprobado(){...}  
}
```

2. Agregar a la clase Nota el método:

```
void recuperar(int nuevoValor){...}
```

3. Implementar la clase Punto. Un Punto en el plano posee coordenada X y coordenada Y. Proporcionar métodos para:

- consultar y cambiar las coordenadas,
- saber si el punto está sobre el eje de las X,
- saber si el punto está sobre el eje de las Y,
- saber si el punto es el origen de coordenadas.

4. Implementar la clase Cubo a partir de la siguiente interfaz:

```
public class Cubo {  
  
    public Cubo (int lado) {...}  
  
    public int medirLongitudLado(){...}  
  
    public void cambiarLongitudLado(int lado) {...}  
  
    public int medirSuperficieCara() {...}  
  
    public void cambiarSuperficieCara(int superficieCara) {...}  
  
    public int medirVolumen() {...}  
  
    public void cambiarVolumen(int volumen) {...}  
}
```

5. Implementar la clase TarjetaBaja a partir de la siguiente declaración:

```
public class TarjetaBaja {  
  
    public TarjetaBaja(double saldoInicial)  
  
    public double obtenerSaldo()  
  
    public void cargar(double monto)  
  
    public void pagarViajeEnColectivo()  
  
    public void pagarViajeEnSubte()  
  
    public int contarViajes()  
  
    public int contarViajesEnColectivo()  
  
    public int contarViajesEnSubte()  
  
}
```

6. Implementar una clase que modele una Circulo, del que se desea conocer: radio, diámetro, perímetro y superficie.

7. Implementar una clase que modele un Disco.

Se desea conocer:

- radio interior
- radio exterior
- perímetro interior
- perímetro exterior
- superficie.

Debe tener operaciones para cambiar el radio interior y el radio exterior.

8. Implementar la clase Ticket a partir de la siguiente interfaz

```
public class Ticket {  
  
    public Ticket()  
  
    public void agregarItem(int cantidad, double precioUnitario)
```

```
public void aplicarDescuento(double porcentaje)

public double calcularSubtotal()

public double calcularTotal()

public int contarProductos()
}
```

9. Implementar la clase CajaDeAhorro con la siguiente interfaz:

```
public class CajaDeAhorro {

    public CajaDeAhorro(String titularDeLaCuenta)

    public String obtenerTitular()

    public double consultarSaldo()

    public void depositar(double monto)

    public void extraer(double monto)
}
```

10. Implementar la clase Cerradura con los siguientes métodos.

```
public class Cerradura {

    public Cerradura(int claveDeApertura, int
cantidadDeFallosConsecutivosQueLaBloquean)

    public boolean abrir(int clave)

    public void cerrar()

    public boolean estaAbierta()

    public boolean estaCerrada()

    public boolean fueBloqueada()

    public int contarAperturasExitosas()

    public int contarAperturasFallidas()
}
```

Considerar que cuando una Cerradura se bloquea no puede volver a abrirse nunca más.

11. Implementar la clase ExpendedorDePasajes con los siguientes métodos.

```

public class ExpendedorDePasajes {
    public ExpendedorDePasajes(double precioPorKm)

    public double venderPasaje(double distanciaEnKm)

    public double venderPasajes(int cantidad, double distanciaEnKm)

    public int pasajesVendidos()

    public double distanciaMaxima()

    public double distanciaPromedio()

    public double ventaTotal()
}

```

12. Implementar la clase Alarma y la clase Sensor con la siguiente interfaz. Una Alarma cuenta con un Sensor de movimiento, un Sensor de contacto y un Sensor de sonido.

```

public class Sensor {

    public void encender()
    public void apagar()
    public boolean activado()
    public void activar()
}

public class Alarma {
    public Alarma(int codigoSeguridad)

    public void encender()

    public void apagar(int codigoSeguridad)

    public boolean activada() {
}

```

Para cada una de las clases construidas en todos los ejercicios, implementar una clase que constituya la prueba de la misma.